REMARKS

Claims 1-57 are pending in the Application. Claims 1-5, 7-10, 22-26, 30-36, 40, 42-47, 49-53, 55 and 56 have been rejected under 35 U.S.C. 102(b) as being anticipated by Kimura et al., U.S. Patent No. 5,546,593. Claims 6, 21, 27-29, 41, 48 and 54 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Kimura in view of Frank et al., U.S. Patent No. 5,790,851.

Applicants note with appreciation that Claims 11-20, 37-39 and 57 have been objected to as being dependent on rejected base claims, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Accordingly, Claim 11 has been canceled and its limitation incorporated into base Claim 1. Similarly, Claim 39 has been cancelled and its limitation incorporated into base Claim 31.

Allowance of base Claims 1 and 31 as amended is therefore respectfully requested. Allowance of dependent Claims 12-20, 57 and 37-39, with amendments to Claims 13 and 18 discussed below, should follow.

Dependent Claims 13 and 18, which formerly depended from Claim 11, have been amended to depend directly from Claim 1.

Claims 22, 24, 26 and 45 have been rejected under 35 U.S.C. 112 second paragraph as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Regarding Claim 22, the Examiner states that there is a typographical error in the identification of the thread of control of which the instructions are fetched.

However, Applicants do not see any typographical error. The term "thread of control" refers to the "thread of control" of Claim 1 (both as originally filed and as amended herein). Although the thread may be halted, it is still possible to fetch the instructions from the halted thread and propagate them into the instruction pipeline, without executing them - thus the thread is "halted". On the other hand, perhaps the Examiner was mistakenly thinking that Claim 22 depends from Claim 21 which refers to "another thread of control," in which case Applicants respectfully draw the Examiner's attention to the fact that Claim 22 does not depend from Claim 21, but rather from Claim 1.

B

Nonetheless, Claim 22 has been amended to recite "fetching instructions from ***said*** thread of control" to clarify that it is the halted thread of control from which instructions are fetched. Furthermore, as there is no antecedent basis for "instruction pipeline", the article has been changed from definite ("the") to indefinite ("a").

Reconsideration and withdrawal of the rejection of Claim 22, in favor of allowance, is respectfully requested, in view of the above changes. If the Examiner continues to reject Claim 22, Applicants request additional specific information as to the offending language.

Regarding Claim 24, the Examiner states that "the program" lacks antecedent basis. However, Applicants respectfully point out that this phrase does not appear in pending Claim 24 which was previously amended in the Preliminary Amendment filed on May 11, 2000. Reconsideration and withdrawal of the rejection of Claim 24 in favor of allowance is thus respectfully requested.

Claim 26 has been amended by changing "program" to "thread of control", for which there is antecedent basis, and should now be allowable. Reconsideration and withdrawal of the rejection of Claim 26 as amended in favor of allowance is thus respectfully requested.

Claim 45, like Claim 26, has been amended by changing "program" to "thread of control", for which there is antecedent basis, and should now be allowable. Reconsideration and withdrawal of the rejection of Claim 45 as amended in favor of allowance is thus respectfully requested.

With regard to the §102 and §103 rejections, none of the cited art discloses an arm instruction as in the present invention. Independent claims 1 and 31 have been argued above and as now amended are believed to be allowable over the cited art. Dependent claims 2-10, 12-30, 32-38, 40-48 and 55-57 should follow. Independent Claims 49 and 50 have now been amended to include an arm instruction and should therefore be allowable over Kimura. Applicants respectfully request allowance of Claims 49 and 50 as amended, as well as dependent Claims 51-54, whose allowance should follow from that of Claim 50.

Accordingly, the present invention as now claimed in claims 1-10, 12-38 and 40-57 is believed to be novel and non-obvious and hence, patentable over the prior art. Allowance is respectfully requested.

B

New Claim 58 incorporates the limitations of base Claim 31 as they existed prior to this amendment, the limitations of intervening Claim 32 and the limitations of Claim 37, as indicated by the Examiner to be allowable. New Claim 59 parallels Claim 38. Allowance of new Claims 58 and 59 is respectfully requested.

New independent Claims 60, 61 and 62 are similar respectively to Claims 1, 31 and 49 as amended, but are intended to cover both situations where the arm instruction identifies the event, as well as situations where event identification is not necessarily done by the arm instruction itself. Allowance is respectfully requested.

Page 4 of the Specification has been amended, as requested by the Examiner, by amending all directly linkable hyperlinks, i.e., by removing "http://" from the referenced URLs. In addition, a typographical error (comma should have been a period) has been corrected in the URL at page 4, line 17.

Page 11, line 21 of the Specification has been amended to correct a typographical error.

No new matter has been added by any of the above amendments.

CONCLUSION

In view of the above amendments and remarks, it is believed that all claims (i.e., Claims 1-10, 12-38 and 40-62) are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned at (978) 341-0036.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By _____

Gerald M. Bluhm
Registration No. 44,035
Telephone: (978) 341-0036
Facsimile: (978) 341-0136

Concord, MA 01742-9133
Dated: 7/11/02

B

MARKED UP VERSION OF AMENDMENTS

Specification Amendments Under 37 C.F.R. § 1.121(b)(1)(iii)

Replace the paragraph at page 4, lines 6 through 10 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph.

One multithreaded computer uses fine-grained multithreading, which is different from SMT, and addresses the synchronization problem with a hardware retry which traps the thread after some number of failures and deschedules it. This is described in "Exploiting Heterogeneous Parallelism on a Multithreaded Multiprocessor," 1992, which can be found at www.tera.com/www/archives/library/psdocs.html [http://www.tera.com/www/archives/library/psdocs.html] .

Replace the paragraph at page 4, lines 16 through 23 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph.

Many papers have been published about Simultaneous Multithreading. For a fairly complete list, see www.cs.washington.edu/research/smt/ [http://www.cs.washington,edu/research/smt/]. The University of Washington has done much work on efficient synchronization on SMT. See, for example, "Supporting Fine-Grained Synchronization on a Simultaneous Multithreading Processor," 1995, available at www.cs.washington.edu/research/smt/papers/hpca.ps [http://www.cs.washington.edu/research/smt/papers/hpca.ps]. A longer version of the paper, UCSD CSE Technical Report #CS98-587, is available at www.cs.washington.edu/research/smt/papers/smt.synch.ps [http://www.cs.washington.edu/research/smt/papers/smt.synch.ps].

B

Replace the paragraph at page 11, lines 20 through 24 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph.

Executing a LDx_ARM on one TPU does not affect any architecturally visible state on another TPU, and in [a] particular cannot clear another TPU's watch_flag, causing the quiescing processor to come out of a quiescent state. Without this restriction, two processors executing LDQ_ARM/QUIESCE sequences could be continually re-arming each other.

Claim Amendments Under 37 C.F.R. § 1.121(c)(1)(ii)

1. (Twice Amended) A method for temporarily halting execution of a thread of control while the thread of control is waiting for an event to occur, comprising:

arming an event monitor, via execution of an arm instruction, by identifying at least one event to be monitored;

requesting that the thread of control be halted until any such identified event is observed by the event monitor; and

if execution of the thread of control has been halted,

monitoring, by the event monitor, for an identified event; and

resuming execution of the thread of control subsequent to observation of an identified event by the event monitor.

13. (Twice Amended) The method of Claim [11] 1, wherein execution of the arm instruction further comprises:

setting an indicator to a first state which enables the event monitor to monitor for the event, wherein the indicator is set to a second state if the event occurs.

18. (Twice Amended) The method of Claim [11] 1, further comprising:

executing a quiesce instruction to request that the thread of control be halted.

B

22. (Twice Amended) The method of Claim 1, further comprising:

while the thread of control is halted,

fetching instructions from [the] said thread of control, and

allowing the fetched instructions to propagate into [the] an instruction pipeline.

26. (Amended) The method of Claim 1 wherein the [program] thread of control is executing in a multiprocessor environment.

31. (Twice Amended) A system for temporarily halting execution of a thread of control while the thread of control is waiting for an event to occur, comprising:

an event monitor which, by execution of an arm instruction, is armed via identification of an event to be monitored; and

an execution scheduler, responsive to the event monitor, which, upon a request that the thread of control be halted until the event is observed by the event monitor, halts execution of the thread of control if the event has not yet occurred since the event monitor was armed, and which resumes execution of the thread of control upon observation of the event by the event monitor.

45. (Amended) The system of Claim 31 wherein the [program] thread of control is executing in a multithreaded environment.

49. (Amended) A system for temporarily halting execution of a thread of control while the thread of control is waiting for an event to occur, comprising:

event monitoring means;

arming means for arming the event monitoring means by identification of at least one event to be monitored, upon the execution of an arm instruction;

requesting means for requesting that the thread of control be halted until any such identified event is observed by the event monitoring means; and

B

halting means for halting the thread of control responsive to the requesting means, wherein if execution of the thread of control is halted, execution of the thread of control is resumed subsequent to observation of an identified event by the event monitoring means.

50. (Amended) An electronic circuit for temporarily halting execution of a thread of control while the thread of control is waiting for an event to occur, comprising:

an event monitor circuit, for monitoring, upon the execution of an arm instruction, for at least one event identified by the thread of control;

a quiesce logic circuit, which, responsive to the event monitor circuit and to a request from the thread of control to quiesce, temporarily halts execution of the thread of control, and which, responsive to the event monitor circuit upon observation of at least one identified event, resumes execution of the halted thread of control.